E815 CLUBS/FNALU User's Guide

Version 1.1

H. Schellman

May 13, 1997

With product testing by: S. Adams, D. Mason, R. Drucker, J. Conrad, J. Goldman, J. Yu, L. de Barbaro and S. Zeller

1 What are FNALU and CLUBS?

1.1 CPU's

Fnalu and CLUBS are central unix systems at fermilab. Fnalu is mainly designed for short jobs which read disk files. Heavy duty access to data tapes is best done through CLUBS which is only accessible through the batch system. fnalu has sgi, IBM, Sun and DEC machines available. CLUBS currently has sgi and IBM machines. To use CLUBS you need to log into one of the IBM or sgi fnalu machines to submit your job.

1.2 Disk

The fnalu/CLUBS system has two kinds of disk.

- AFS disk - your user account will be in AFS space. The good news is that it is backed up and you get very fast connections even if you are not on the same machine as the disk. The bad news is that you will get 200 MB if you're lucky and your disk will have kerberos security which adds a new system to learn and makes it hard to access from remote machines through rsh.
- local disk - E815 has gotten about 21 GB of local disk for the group. This disk is only directly accessible from the fnalu nodes. We will be staging data onto this disk. This disk can not be seen directly from the CLUBS system.

1.3 Tape access

There are two kinds of tape access available.

- 8mm tapes can be read through the `needfile` command which stages the files into the robot and then, on demand, stages them to a temporary disk for you to use. Once files are in the robot, they will stay there for weeks and can be accessed within a few minutes. The command `ndflist` tell you which tapes have been staged into the robot.

  8mm output is done through the reverse process. Because files are stored temporarily in the robot, you can do many streams and then sort them when you dump from the robot back to 8mm.

- One can also get permanent robot space. We have not gotten this set up yet but it seems a great way to get at our data. Robot access speeds are generally 5-10 minutes for any 150MB file you might wish.

1.4 Batch systems

fnalu/CLUBS have moved over to a new batch system called `lsf`. For backwards compatibility, it is also called `fbatch`. It has very nice `man` pages which list the useful commands. The batch system tries to balance the load between various systems and users in a fair way.

CLUBS is only accessible through the batch system.

On fnalu you can either run interactive or batch but for any CPU utilization beyond 15 minutes, batch is strongly suggested. The system will `nice` longer interactive jobs down to zero and at some point, the system operators will turn you in.

1.5 How CLUBS works

Clubs is a remote system which is mainly used for tape access. You submit a job to it and it gives you about 1GB of temporary disk space. Files can be staged from the robot to this temporary space, processed and then deleted. The current version of the cruncher knows how to read tapes in this way. The tape writing system was recently changed and needs to be reimplemented in the cruncher.

Because the tape is copied to local disk for processing, everyone can run on the same tape simultaneously. In fact, popular tapes will usually have been prestaged by someone else and be ready to go when you want them.

2 How do I get an account

1. If you do not already have a central system account you need to put in the standard FNAL account request form with Bob or Mike's signature.
2. Many people already have fnalu accounts but not CLUBS. If that is true or if you have forgotten your password, contact Yolanda Valadez (COMPDIV@FNAL) and ask her for help.
3. Once you have your account, you use `kpasswd` to change passwords.
4. You may not have proper `.login` and `.cshrc` files. Copy them from the e815 account on fnalu.

   This login, among other things, does the setups needed to run batch jobs.

   `setup fbatch` sets up the batch system

   `setup nt` sets up some useful utilities. WARNING - a very nasty feature is that it sets the pointer for temporary space to your local (small) directory. You must issue the command:

   `setenv $FERMINT_DPOOL_DIR`

   after `setup nt` to clear this default setting or you will find your home directory full of large data files. The e815 login does this by default.
5. You will need to set up products like emacs yourself as it is not done in the default login.

At this point you should be ready to go.

3 Running E815 jobs on fnalu/CLUBS

Only crunch jobs have been implemented as they require tape reading. Other executables should run very easily.

3.1 Prestaging a tape

You can't really run on a tape until it has been read into the robot, partially because the information about number of files is not available. To get a tape into the robot issue the command:

`~e815/bin/prestage VSN`

where VSN is the 6-character tape id. It will start prestaging, you can use `ndflist` to follow the progress of the prestage. You can access files once they have been prestaged. Submitting a batch job will also do this for you but unless you know how many files are on the tape, you can get in trouble.

3.2 Running a crunch job

Right now, we do not have CVS implemented on the system. To run a job, you will need to set up a local project directory, do some local setup, `ftp` across `nucrunch.crd`, an executable and the `data` subdiretory. I have set up a script which will do the most simple and useful operation on a tape, allowing you to read files within a certain range or the whole tape. Other scripts, to read multiple tapes or read disk can be set up by analogy.

What the script does is take simple information from the command line and the file `job.def` and use it to build two scripts, `jobname.cmd` and `jobname.lsf`. `jobname.cmd` is a csh script which clubs will be asked to execute and `jobname.lsf` is the one line command which submits the job.

Here is an itemized procedure.

1. make and test an executable on a short disk file on the e815 sgi's
2. do a `make clubs` to make the CLUBS executable `crnch_clubs`
3. log onto fsgi02.
4. create a master subdirectory for your project
5. while in that subdirectory type `~e815/bin/e815_setup normal` - this will copy over some useful files and create a data subdirectory for you. Testbeam users and others will need to specify more directories.
6. Edit the file `job.def` to notify you, not me, when your job starts.

7. `ftp` over `crnch_clubs`, `nucrunch.crd` and the `data` subdirectory from the e815 cluster. Remember to type `lcd data` when you get to that point or your data files will end up in the wrong place. Once again, testbeam files will need to be moved over separately for now.

8. change `crnch_clubs` so that it is executable by doing a `chmod +x crnch_clubs`

9. edit `nucrunch.crd` to satisfy your new needs: on clubs you can either read a disk file or from input lists. To use the default input list set up by this example script you must get rid of `TYPE RTAP`, `TYPE RDSK` and put in the line

   `TYPE LSTD list.inp`

   Do not use the default `LSTD data/list.inp` as the `data` area is set up as a link to save space.

   Note, these are limitations of the specific script I have set up. Running off of disk is actually easier and you can also specify your own input list instead of having my script build it.

10. type ~`e815/bin/submit_job TAPE first last queue`.

    This will create a special subdirectory `TAPE_first_last` with a COPY of `nucrunch.crd` and LINKS to the executable and data areas. This allows you to run multiple jobs with the same executable and data. If you want different data/executables, making another master project directory is suggested.

    (a) `TAPE` is the tape you want to read

    (b) `first` is the first file on the tape you wish to read

    (c) `last` is the last file on the tape you wish to read

    If `first` and `last` are set to 0 0, the script will try to find the last file on the tape by calling the program `ndflist`. If the tape has been prestaged, this information will be available. Otherwise the script will start a prestage, tell you it has done so and quit.

    The script will use the `first` and `last` information to create an input list for you. It will be stored as `list.inp` not the default `data/list.inp`. If you wish to make your own input list, just change `nucrunch.crd` to point to it instead of `list.inp`.

(d) queues are `30min`, `4hr`, `12hr`, `1day`. These are abbreviations for the clubs queues `30min_disk` ...

This script will set up the job in the subdirectory `TAPE_first_last`. It makes links to the executable and data areas, makes a copy of `nucrunch.crd`. It also creates the file `TAPE_first_last.cmd` which is the script the batch processor will run, and the file `TAPE_first_last.lsf` which is the command to the batch system telling it to run the job. The last thing the script does is execute `TAPE_first_last.lsf` to start the job. It will ask for your password so that clubs has access to your disk.

4 What `submit_job` does

Here is an example submission. My project directory needs to look like this before I start.

```
-rw-r--r--    1 schellma e665       52 May  9 17:17 job.def
drwxr-xr-x    2 schellma e665     2048 May  9 17:19 data
-rwxr-xr-x    1 schellma e665  3618904 May  9 17:23 crnch_clubs
-rw-r--r--    1 schellma e665     2568 May 11 11:04 nucrunch.crd
```

Now I do a job submission to read the first 4 tapes on GO0514.

```
fsgi02:: #37 -> ~e815/bin/submit_job GO0514 1 4 4hr

 this job will read these files
scratch_dir/GO0514.01
scratch_dir/GO0514.02
scratch_dir/GO0514.03
scratch_dir/GO0514.04
 now submitting the job - will ask for your password
```

```
Enter AFS password:
fbatch_sub executing LSF command locally on fsgi02....
Needfile statements found, submitting job on hold...
Job <30887> is being stopped
Parameters of job <30887> are being changed
Job successfully suspended, calling the prestager...
request (LSF.30887) will be prestaged
Prestager successfully called...
fsgi02:: #38 ->
```

These messages indicate that the script has been successful in setting up the job, submitting it and in telling the prestager it needs the tape. The job number 30887 is useful if you want to track or kill the job.

4.1 What is going on when the `submit_job` script runs

it has created a subdirectory just for this job called `GO0514_1_4` and put the files needed to run the job in that directory.

Contents of `GO0514_1_4`

```
ls -lrt GO0514_1_4
total 14
-rw-r--r--    1 schellma e665   140 May 13 20:35 stripper.outlist
-rw-r--r--    1 schellma e665    88 May 13 20:35 list.inp
-rw-r--r--    1 schellma e665  2568 May 13 20:35 nucrunch.crd
-rwxr-xr-x    1 schellma e665   405 May 13 20:35 GO0514_1_4.cmd
lrwxr-xr-x    1 schellma e665     7 May 13 20:35 data -> ../data
-rwxr-xr-x    1 schellma e665   138 May 13 20:35 GO0514_1_4.lsf
lrwxr-xr-x    1 schellma e665    14 May 13 20:35 crnch_clubs -> ../crnch_clubs
lrwxr-xr-x    1 schellma e665    10 May 13 20:35 job.def -> ../job.def
lrwxr-xr-x    1 schellma e665    21 May 13 20:36 scratch_dir -> /tmp/spacall.pid24
-rw-r--r--    1 schellma e665     0 May 13 20:40 go0514.01.hst
```

note that the `data` and `crnch_clubs` are links while `nucrunch.crd` and `list.inp` have been copied to this subdirectory. This lets you run multiple jobs with different input streams but the same code and constants.

This job has started running so it has also opened a histogram file and had it's scratch area defined.

When it is done, output files `GO0514_1_4.out` and `GO0514_1_4.err`, and any others you create, will appear in this directory.

4.2 jobname.cmd

This is the script which is sent to clubs to run.

```
#!/bin/csh -f
#
## prestage the tape
needfile -vsn GO0514 -tape 8mmd -file 1 -noaccess -lrecl 7200 -block 7200
#
## make a link to clubs scratch so you don't have to use environmentals
ln -s $FERMINT_DPOOL_DIR scratch_dir
touch scratch_dir/checkfile
## begin file 1  by staging it
needfile -vsn GO0514 -tape 8mmd -lb GO0514.01 -file 01 -lrecl 7200 -block 7200
ls -lrt scratch_dir
date
crnch_clubs
date
ls -lrt scratch_dir
```

Most of the complication here is that one needs to prestage the tape and then to stage the first file. After that, all you need to do is run the executable.

To run a job which just reads disk, you could have a cmd file as simple as:

```
#!/bin/csh -f
crnch_clubs
```

which is why I haven't written a special script for you.

4.3 jobname.lsf

This is the command that starts the job:

```
fbatch_sub   -u  "schellman@fnal.gov"  -N -B  -o GO0514_1_4.out -e GO0514_1_4.err\
-q 30min_disk  -J GO0514_1_4  -R "irix"  GO0514_1_1.cmd
```

To submit a job without reading tape, you only have to change the queue from `30min_disk` to `30min` (and could just leave it as is, you don't need the disk but it won't hurt you).

4.4 What clubs looks for in a list.inp file

Clubs has assigned the temporary disk that it got to the system to the directory `$FERMINT_DPOOL_DIR`, our script has set up a link from the directory `scratch_dir` to this directory. The data staged from the robot will appear in this directory and `list.inp` tells it to look there.

```
scratch_dir/GO0514.01
scratch_dir/GO0514.02
scratch_dir/GO0514.03
scratch_dir/GO0514.04
```

the `clubs` option in your `make` fired up code which opens one of these files, then asks the robot to feed it the next file. When a file has been read, it is deleted to make room for the next one. All of this is done in fortran calls in the routine `process.f` which were not easy to figure out.

5 The `fbatch` system

`fbatch` is a fermilab pseudonym for the batch system `lsf`. It was installed about 4 months ago and is still being tuned but it is much, much better than the old system.

To learn about `fbatch`, check out the web pages or type `man fbatch`, which lists the commands, or `man fbatch command` for details.

I have found the following commands very useful

5.1 fbatch_submit

This is the command used in `submit_job` to start the job. It has many useful arguments including the routing for your standard output and error files.

It also sets up the queues, which are:

`30min`, `4hr`, `12hr`, `1day`, `4days` for non-clubs jobs

`30min_disk`, `4hr_disk`, `12hr_disk`, `1day_disk`, `4days_disk` for clubs jobs.

disk means you need temporary cache space.

5.2 fbatch_jobs

This tells you which jobs are running and what their state is:

`PEND` means waiting for a prestage
`RUN` means running
`PSUSP` means suspended due to system load

the option `-u` all shows you everyone's jobs so you can whine about the queuing system.

The `-l` option gives you more information.

5.3 fbatch_queues

This will list the available queues. The `-l` option gives you extra information.

5.4 fbatch_kill

Each job is assigned a job number which you can find via `fbatch_jobs`. To kill type `fbatch_kill nnnnn`.

5.5 fbatch_stop

This will suspend a job for you.

5.6 fbatch_resume

This will resume a suspended job.

6 needfile and ndflist

`needfile` is the command you use to ask the robot to give you a file. It can be called from the command line or from fortran.

`ndflist` is the command which tells you if a tape has been prestaged yet and how many files are on it.

```
ndflist -i GO0514

================================================
GO0514    6 files  811 MB
File           size location
----------------------------
   1   157284000 TAPE-caphsm1
   2   157284000 TAPE-caphsm1
   3   157284000 TAPE-caphsm1
   4   157284000
   5   157284000
   6    62661600
```

This shows that the tape GO0514 has 6 files and that, at least the 1st 3 are tape in the robot.

7 What secret special AFS features do I need to know

7.1 Token timeouts

As part of the enhanced security, your password is used to set up something called a token which allows you to access disk. These tokens have a default

lifetime of 25 hours. After that point you lose write access to your disk unless you revive it by typing `klog`, which will ask you to renter your password.

Clubs jobs need a token to write to your local disks. They will ask for you password and set one up at start time. But 25 hours later, the token disappears and your job dies. This is usually not a problem but, if you are running 12 hour jobs on clubs, you need to request an increase in token lifetime.

7.2 Disk quotas

AFS uses local caching to bring copies of your centrally stored files to the local machine. This gets rid of the annoying network problems found on the local cluster but means a `df` doesn't work to find your quota. The command

```
fs listquota
```

will tell you how much space you have left.

7.3 File protection

Warning ! AFS is really cool, it makes a world-wide directory tree which includes SLAC, DESY, CERN. The bad news is that all of these people can read your files unless you set the access codes by using the command which I have to remember. E815 users are strongly urged to only use this feature to protect their email and sensitive personnel items. Making your analysis unavailable to your collaborators is not nice.

7.4 Long paths

Because AFS is worldwide, path names look like:

`afs/fnal/files/home1/schellma`....

This adds a lot of string overhead and often overflows if you set up feature-laden products like `exmh` and `tktools` at the same time. What I do is `setup exmh`, `exmh &`, `unsetup exmh`, `setup tktools` to get around this problem.

8 Useful documents

The following web pages are useful:

accounts - http://www.fnal.gov/cd/main/accounts.html
AFS - http://www-oss.fnal.gov/hanson/afs.html
applications - http://www-oss.fnal.gov/uas/
batch systems - http://www-hppc.fnal.gov/batch/batch.html
cernlib - http://fnpspa.fnal.gov/cern.html
fbatch itself - http://fnhppc.fnal.gov/fbatch/fbatch.html
fnalu - http://www-oss.fnal.gov/hanson/afs/fnalu_ug.ps
fmms robot - http://www-hppc.fnal.gov/mss/fmss.html
OCS - http://www-oss.fnal.gov/hanson/ocs.ps
people - http://www-oss.fnal.gov/dept/org/index.html
printing - http://www.fnal.gov/cd/main/printing.html
unix at fnal - http://www.fnal.gov/cd/UNIX/UnixResources.html